

Übung zur Lehrveranstaltung

Semantic Web Technologies 1

Andreas Harth und Anees Mehdi
Wintersemester 2013/14
<http://semantic-web-grundlagen.de>

Übung 2: Logik und RDF-Semantik

Hinweis zur nächsten Aufgabe: Eine *Teilformel* einer logischen Formel ist jeder Teilabschnitt der Formel, der selbst eine wohlgeformte logische Formel darstellt, wobei die vollständige Klammerung beachtet wird. Zum Beispiel sind die Teilformeln von $(q \rightarrow ((p \wedge q) \wedge r))$ genau die Formeln $(q \rightarrow ((p \wedge q) \wedge r))$, $((p \wedge q) \wedge r)$, $(p \wedge q)$, p , q , r . Dagegen ist z.B. $(q \wedge r)$ keine Teilformel (wegen der Klammerung). Um den Wahrheitswert einer aussagenlogischen Formel unter einer Interpretation \mathcal{I} zu berechnen, bestimmt man schrittweise die Werte für alle Teilformeln, wobei die Werte für die kleinsten Teilformeln (d.h. für einzelne Propositionen wie p oder q) direkt durch die Interpretation \mathcal{I} vorgegeben wird.

Aufgabe 2.1 Entscheiden Sie für jede der folgenden Formeln, ob diese allgemeingültig, erfüllbar, widerlegbar, oder unerfüllbar ist. Beweisen Sie Ihre Antwort jeweils durch Angabe einer Wahrheitstafel, in der die Werte für jede Teilformel (und die gesamte Formel) jeweils in einer Spalte dargestellt werden.

Beispiel: Die Formel $(q \rightarrow (p \wedge q))$ ist erfüllbar und widerlegbar. Die entsprechende Wahrheitstafel ist: (wir schreiben hier $\mathcal{I}(\cdot)$, um darzustellen, dass die angegebenen Werte erst durch Anwendung einer bestimmten Interpretation \mathcal{I} festgelegt werden):

$\mathcal{I}(p)$	$\mathcal{I}(q)$	$\mathcal{I}((p \wedge q))$	$\mathcal{I}((q \rightarrow (p \wedge q)))$
t	t	t	t
t	f	f	t
f	t	f	f
f	f	f	t

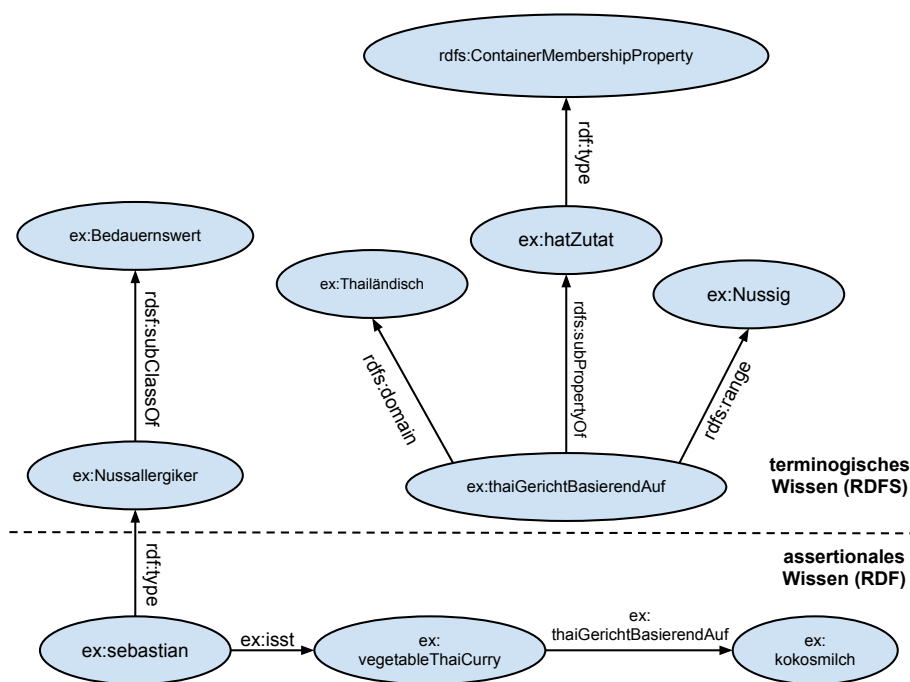
- (a) $(p \vee \neg p)$
- (b) $((p \vee q) \rightarrow (\neg p \vee \neg q))$
- (c) $\neg((p \rightarrow q) \leftrightarrow (\neg p \vee q))$
- (d) $((p \rightarrow q) \rightarrow p) \rightarrow p$
- (e) $((p \wedge q) \rightarrow r) \leftrightarrow (p \rightarrow (q \rightarrow r))$
- (f) $((p \wedge \neg p) \rightarrow q)$

Aufgabe 2.2 Verdeutlichen Sie sich die Begriffe „Theorie“, „logische Konsequenz“ und „Äquivalenz“ und entscheiden Sie dann, ob folgende Behauptungen richtig sind. Geben Sie jeweils eine (informelle) Begründung für Ihre Antwort.

Für beliebige Theorien \mathcal{T} und \mathcal{S} gilt:

- (a) Ist eine Formel F allgemeingültig, dann gilt $\mathcal{T} \models F$, d.h. aus jeder Theorie folgen zumindest alle Tautologien.
- (b) Je größer eine logische Theorie ist, desto mehr Modelle hat sie. Das heißt, wenn $\mathcal{T} \subseteq \mathcal{S}$, dann ist jedes Modell von \mathcal{T} auch ein Modell von \mathcal{S} .
- (c) Je größer eine Theorie ist, desto mehr logische Konsequenzen hat sie. Das heißt, wenn $\mathcal{T} \subseteq \mathcal{S}$, dann ist jede logische Konsequenz aus \mathcal{T} auch eine Konsequenz aus \mathcal{S} .
- (d) Ist $\neg F \in \mathcal{T}$, dann kann $\mathcal{T} \models F$ niemals gelten (wobei F eine beliebige Formel ist).
- (e) Sind zwei Theorien unterschiedlich ($\mathcal{T} \neq \mathcal{S}$), dann unterscheiden sie sich auch in wenigstens einer logischen Konsequenz (zum Beispiel, indem es eine Formel F gibt, so dass $\mathcal{T} \models F$ aber $\mathcal{S} \not\models F$).

Aufgabe 2.3 Beschreiben Sie eine RDFS-Interpretation, die Modell der Beispiel-Ontologie aus Übung 1.4 ist. Zur Erinnerung hier noch einmal die Darstellung dieser Ontologie als RDFS-Graph:



Aufgabe 2.4 Nennen Sie für die Ontologie aus Aufgabe 2.3

- ein Tripel, welches einfach folgt,
- ein Tripel, welches RDF-folgt, aber nicht einfach folgt,
- ein Tripel, welches RDFS-folgt, jedoch nicht RDF-folgt.

Aufgabe 2.5 Wie Sie wissen, gilt in RDF(S) die *unique name assumption* nicht, das heißt, es ist möglich, dass in einem Modell mehreren unterschiedlichen URIs dieselbe Ressource zugeordnet wird. Überlegen Sie, ob (und gegebenenfalls wie) es sogar möglich ist, in RDFS zu spezifizieren, dass zwei gegebene URIs dieselbe Ressource bezeichnen.

Aufgabe 2.6 Der *leere Graph* enthält keine Tripel (entspricht also der leeren Menge). Zeigen Sie durch Ableitungen, dass die folgenden Tripel aus dem leeren Graphen RDFS-folgen:

- (a) `rdfs:Resource rdf:type rdfs:Class .`
- (b) `rdfs:Class rdf:type rdfs:Class .`
- (c) `rdfs:Literal rdf:type rdfs:Class .`
- (d) `rdf:XMLLiteral rdf:type rdfs:Class .`
- (e) `rdfs:Datatype rdf:type rdfs:Class .`
- (f) `rdf:Seq rdf:type rdfs:Class .`
- (g) `rdf:Bag rdf:type rdfs:Class .`
- (h) `rdf:Alt rdf:type rdfs:Class .`
- (i) `rdfs:Container rdf:type rdfs:Class .`
- (j) `rdf:List rdf:type rdfs:Class .`
- (k) `rdfs:ContainerMembershipProperty rdf:type rdfs:Class .`
- (l) `rdf:Property rdf:type rdfs:Class .`
- (m) `rdf:Statement rdf:type rdfs:Class .`
- (n) `rdfs:domain rdf:type rdf:Property .`
- (o) `rdfs:range rdf:type rdf:Property .`
- (p) `rdfs:subPropertyOf rdf:type rdf:Property .`
- (q) `rdfs:subClassOf rdf:type rdf:Property .`
- (r) `rdfs:member rdf:type rdf:Property .`
- (s) `rdfs:seeAlso rdf:type rdf:Property .`
- (t) `rdfs:isDefinedBy rdf:type rdf:Property .`
- (u) `rdfs:comment rdf:type rdf:Property .`
- (v) `rdfs:label rdf:type rdf:Property .`